

## P1

(1) The *DrawBezier.m* uses the de Casteljau subdivision method and plots a cubic spline specified by a sequence of de Boor controls points from the user. The variable  $n$  is the number of iterations and  $t$  is the size of the subdivision. Here,  $n$  will always be 6 in order to draw a smooth line, and  $t$  is 0.5. The user presses enter to indicate all the desired points have been selected.

```
n = 6;
t = 1/2;
[x,y] = ginput();
d = [x,y]; % d(i,:) i = 1,2,... represents a point
d = sortrows(d); % sort our data
b = calculateDE(d, n, t); % calculate the points used to draw
the curve
b = sortrows(b); % sort our data
plot(d(:,1), d(:,2)); % draw the input data d
hold on;
plot(b(:,1), b(:,2), 'b-') % draw our curve
title('Bezier_Curve')
```

## P2

(2) The *drawDE.m* uses the de Casteljau subdivision method and yields a polygonal line which approximates *Bezier curve*. The variable  $n$  is the number of iterations and  $t$  is the size of the subdivision. The user presses enter to indicate all the desired points have been selected.

For example if we input  $n = 5$ ,  $t = \frac{1}{2}$  and a series points through screen:

```
n = 5;
t = 1/2;
%% User input of the data
[x,y] = ginput();
d = [x,y]; % d(i,:) i = 1,2,... represents a point
d = sortrows(d); % sort our data
%% Run the subdivision and draw curve
b = calculateDE(d, n, t);
```

```

% calculate the points used to draw the curve
b = sortrows(b); % sort our data
plot(d(:,1), d(:,2), 'r*'); % draw the input data d
hold on;
plot(b(:,1), b(:,2), 'b-') % draw our curve
title('Bezier_Curve')

```

Then we can get the graph:

